Addition is exponentially harder than counting for shallow monotone circuits

Xi Chen* Igor C. Oliveira† Rocco A. Servedio‡

Department of Computer Science
Columbia University

September 7, 2018

Abstract

Let $U_{k,N}$ denote the Boolean function which takes as input k strings of N bits each, representing k numbers $a^{(1)}, \ldots, a^{(k)}$ in $\{0, 1, \ldots, 2^N - 1\}$, and outputs 1 if and only if $a^{(1)} + \cdots + a^{(k)} \geq 2^N$. Let $\mathsf{THR}_{t,n}$ denote a *monotone unweighted threshold gate*, i.e., the Boolean function which takes as input a single string $x \in \{0,1\}^n$ and outputs 1 if and only if $x_1 + \cdots + x_n \geq t$. The function $U_{k,N}$ may be viewed as a monotone function that performs addition, and $\mathsf{THR}_{t,n}$ may be viewed as a monotone function that performs counting. We refer to circuits that are composed of THR gates as *monotone majority circuits.*

The main result of this paper is an exponential lower bound on the size of bounded-depth monotone majority circuits that compute $U_{k,N}$. More precisely, we show that for any constant $d \geq 2$, any depth-d monotone majority circuit computing $U_{d,N}$ must have size $2^{\Omega(N^{1/d})}$. Since $U_{k,N}$ can be computed by a single monotone *weighted* threshold gate (that uses exponentially large weights), our lower bound implies that constant-depth monotone majority circuits require exponential size to simulate monotone weighted threshold gates. This answers a question posed by Goldmann and Karpinski (STOC'93) and recently restated by Håstad (2010, 2014). We also show that our lower bound is essentially best possible, by constructing a depth-d, size-$2^{O(N^{1/d})}$ monotone majority circuit for $U_{d,N}$.

As a corollary of our lower bound, we significantly strengthen a classical theorem in circuit complexity due to Ajtai and Gurevich (JACM'87). They exhibited a monotone function that is in AC^0 but requires super-polynomial size for any constant-depth monotone circuit composed of unbounded fan-in AND and OR gates. We describe a monotone function that is in depth-3 AC^0 but requires *exponential* size monotone circuits of any constant depth, even if the circuits are composed of THR gates.

*xichen@cs.columbia.edu.
†oliveira@cs.columbia.edu.
‡rocco@cs.columbia.edu. Supported in part by NSF grants CCF-1319788 and CCF-1420349.

1 Introduction.

> *"And you do Addition?" the White Queen asked. "What's one and one and one and one and one and one and one and one and one and one?"*
>
> *"I don't know," said Alice. "I lost count."*
>
> *"She can't do Addition," the Red Queen interrupted.*
>
> — Lewis Carroll, *Through the Looking Glass*

Threshold functions and threshold circuits. A Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is called a *weighted threshold function* (also known as a halfspace, weighted majority, weighted threshold gate, or linear threshold function) if there exist integers $w_1, \ldots, w_n$ and t such that

$$f(x) = 1 \quad \Longleftrightarrow \quad \sum_{i=1}^{n} w_i x_i \geq t.$$

The parameters $w_1, \ldots, w_n$ are called *weights*. We say that a threshold function f is *unweighted* if $|w_i| = 1$ for every $i \in \{1, \ldots, n\}$, and that it is *monotone* if every weight is non-negative. (Thus a monotone unweighted threshold function is precisely a $\mathsf{THR}_{t,n}$ function described in the abstract.)

Threshold functions and their generalizations have been extensively investigated for decades (see e.g. Dertouzos [Der65], Minsky and Papert [MP68], and Muroga [Mur71]), and arise in diverse areas including social choice theory (Taylor and Zwicker [TZ92]), circuit complexity (Aspnes et al. [ABFR94]), structural complexity (Beigel, Reingold, and Spielman [BRS95]), learning theory (Freund and Schapire [FS97]), neural networks (Parberry [Par94]), cryptography (Naor and Reingold [NR04]), and many others.

In this work, we consider Boolean circuits that are composed of gates that compute threshold functions (i.e., *threshold gates*). (We refer to Jukna [Juk12] as an extensive reference on Boolean functions and circuit complexity). While individual threshold gates may appear relatively simple, Boolean circuits composed of these gates (i.e., *threshold circuits*) remain poorly understood despite intensive study. For instance, it is a notorious and long-standing open problem in complexity theory to prove the existence of a function in NP that cannot be computed by a depth-2 circuit with polynomially many weighted threshold gates. This difficulty can be explained in part by the surprising computational power of bounded-depth threshold circuits, both in theory and practice. On the theory side, such circuits can efficiently implement all the basic arithmetic operations (see e.g., Table 1 in Sherstov [She07]) and can also simulate (in quasi-polynomial size and depth 3) $\mathsf{AND/OR/MOD}_m$ Boolean circuits of much larger depth (Allender [All89] and Yao [Yao90]). On a more practical level, constant-depth networks of (continuous analogues of) threshold gates play a fundamental role in recent successful deep learning frameworks (see e.g., Schmidhuber [Sch15]).

Despite our inability to prove strong lower bounds against threshold circuits, there have been some notable successes in understanding the relative power of weighted versus unweighted threshold gates and circuits. Siu and Bruck [SB91] were the first to show that any weighted threshold gate can be simulated by a polynomial-size, constant-depth circuit consisting of unweighted threshold gates (such circuits are also known as *majority circuits*). This result was improved by Goldmann, Håstad, and Razborov in [GHR92], who showed (non-constructively) that weighted threshold gates can be computed by polynomial-size majority circuits of depth 2; in fact, [GHR92] showed that any

depth-d weighted threshold circuit can be simulated efficiently by a depth-$(d+1)$ majority circuit. Soon thereafter Goldmann and Karpinski [GK93] gave a constructive proof with better parameters for the size of the resulting majority circuits. Subsequent simplifications and improvements of these simulations were given by Hofmeister [Hof96] and Amano and Maruoka [AM05].

Monotone functions and monotone circuits. In a different, and highly successful, strand of circuit complexity research, a wide range of lower bounds have been obtained against various types of *monotone* Boolean circuits (composed of AND/OR gates only but no negations). A sequence of well-known results [Raz85, And85, AB87, Tar88] culminated in the existence of explicit monotone Boolean functions that can be computed by polynomial-size Boolean circuits but require *monotone* circuits of exponential size. Analogous results highlighting the limitations of monotone circuits are also known at the "low-complexity" end of the spectrum: in an important result, Ajtai and Gurevich [AG87] exhibited a monotone function in AC^0 (i.e., a constant-depth, polynomial-size AND/OR/NOT Boolean circuit) that requires *monotone* AC^0 circuits (composed of AND/OR gates) to have super-polynomial size. However, it should be noted that the Ajtai–Gurevich circuit lower bound against monotone AC^0 is quantitatively not very strong (at best a quasipolynomial $n^{\Omega(\log n)}$ lower bound; see discussion following the statement of the Ajtai–Gurevich theorem below). Other works have given alternative/simplified expositions of the Ajtai–Gurevich lower bound and of its consequences in formal logic (see [BST13] for the former and Stolboushkin [Sto95] for the latter). But prior to the results of this paper, stronger lower bounds against monotone AC^0 circuits for monotone functions in AC^0 remained elusive.

This work: Monotone weighted threshold functions versus constant-depth monotone majority circuits. As mentioned earlier, Goldmann and Karpinski gave a constructive proof [GK93] that weighted threshold gates can be simulated by polynomial-size and depth-2 majority circuits. They also observed that even if the weighted threshold gate is monotone, known simulations produce majority circuits that are inherently *non-monotone* (i.e., they contain majority gates with negative weights, or equivalently, negation gates), which then led them to ask the question of whether an efficient monotone simulation is possible in constant depth.

Hofmeister [Hof92] made some early progress on this question by showing that any monotone depth-2 majority circuit that computes the function $U_{2,N}$ from the abstract must have exponential size. To state the result more precisely, let us first clearly specify our notion of monotone majority circuits. A monotone majority circuit here is a directed acyclic graph which may have multiple edges (called wires). There is a single node with no outgoing wires, called the output gate. Nodes that have no incoming wires are called input nodes and are each labeled either 0, 1 or x_i, for some i; every other node is labeled with a monotone unweighted threshold gate $\mathsf{THR}_{t,m}$ for some t, with m being its in-degree, which outputs 1 iff there are at least t 1's from its m input wires. We say the *size* of a monotone unweighted threshold gate $\mathsf{THR}_{t,m}$ is m (or its in-degree), and that the size of a monotone majority circuit is the sum of the sizes of its gates (or its number of wires).[1] Then Hofmeister showed that every depth-2 monotone majority circuit for $U_{2,N}$ must have size $2^{\Omega(\sqrt{N})}$.

As mentioned above, in subsequent work [Hof96] and [AM05], several improvements were made on the Goldmann-Karpinski simulation, but neither is monotone, and no further progress was obtained on the lower bound side after Hofmeister's paper [Hof92] until the current work. The question

[1] Observe that by reduplicating inputs, any weighted threshold function f given by $\sum_{i=1}^{n} w_i x_i \geq t$ can be computed by an unweighted threshold gate of size $|w_1| + \cdots + |w_n|$. We sometimes refer to this as the "weight of f."

of Goldmann and Karpinski was recently restated by Håstad [Hås10, BHKS14].

1.1 Our Results.

Our main result shows that monotone weighted threshold gates cannot be simulated by subexponential size monotone majority circuits of constant depth. This may be viewed as an extension of Hofmeister's depth-2 lower bound in [Hof92] to arbitrary constant depth (in fact we obtain superpolynomial size lower bounds even for circuits of small super-constant depth; see discussions after Theorem 1 below). We thus answer the question posed by Goldmann and Karpinski [GK93] and by Håstad [Hås10, BHKS14].

Before giving a precise statement of our results, we define formally the family $U_{k,N}$ of Boolean functions as described in the abstract. Given $t \geq 1$, we let $[t]$ denote the set $\{1, \ldots, t\}$. For $k \geq 2$, the function $U_{k,N}$ maps $\{0,1\}^{k \times N}$ to $\{0,1\}$ as follows. Given $x = (x_{i,j})_{i \in [k], j \in [N]} \in \{0,1\}^{k \times N}$, define

$$\mathsf{SUM}(x) \stackrel{\text{def}}{=} \sum_{j=1}^{N} 2^{N-j} \cdot (x_{1,j} + \cdots + x_{k,j}) \qquad \text{and} \qquad U_{k,N}(x) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \mathsf{SUM}(x) \geq 2^N, \\ 0 & \text{otherwise.} \end{cases}$$

It is helpful to think of the input $x = (x_{i,j})_{i \in [k], j \in [N]}$ as a k-row, N-column, and 0/1-valued matrix, where its ith row $(x_{i,1}, \ldots, x_{i,N})$ gives the binary representation of a number $x^{(i)} \in \{0, 1, \ldots, 2^N - 1\}$ in the usual way (with $x_{i,1}$ being the most significant bit). Then the function $U_{k,N}$ adds up the k numbers $x^{(1)}, \ldots, x^{(k)}$ and outputs 1 if and only if the sum is at least 2^N.

With the definition of $U_{k,N}$ in place, our main result can be stated as follows:

Theorem 1. *Let $d \geq 2$, n and N be three positive integers that satisfy*

$$n \geq 2^{60d} \quad \text{and} \quad N \geq (2^{13} n)^d.$$

Then any depth-d monotone majority circuit that computes $U_{d,N}$ must have size at least $2^{n/2^{60d}}$.

This lower bound is nearly optimal for any fixed $d \geq 2$, as we prove the following upper bound.

Theorem 2. *Let $k, d, N \geq 2$ be three positive integers. Then there exists a depth-d monotone majority circuit of size $2^{6(N^{1/d} \log k + \log N)}$ that computes $U_{k,N}$.*

Remark 1. For any fixed constant $d \geq 2$, Theorems 1 and 2 together show that the smallest depth-d monotone majority circuit that computes $U_{d,N}$ (note that this function has $d \cdot N = \Theta(N)$ input variables) has size $\exp(\Theta(N^{1/d}))$. In addition, by setting $d = c\sqrt{\log N}$ and $n = 2^{61d}$ for some small enough positive constant c so that $N \geq (2^{13} n)^d$, Theorem 1 implies that any depth-d monotone majority circuit computing $U_{d,N}$ has superpolynomial size (exponential in $2^{c\sqrt{\log N}}$).

Remark 2. As an easy consequence of Theorem 2, we obtain a slightly weaker version of the main result of Beimel and Weinreb [BW05]. They proved that the "universal monotone threshold function"[2] $U_{O(N), O(N \log N)}$ can be computed by a poly(N)-size, depth-$O(\log N)$ monotone circuit composed of fan-in two AND gates and unbounded fan-in OR gates. While Theorem 2 above is tailored for small values of k, we note that it implies that $U_{O(N), O(N \log N)}$ can be computed by a

[2] It is called the universal monotone threshold function because it can simulate any monotone weighted threshold function over N inputs.

poly(N)-size, depth-$O(\log^2 N)$ monotone circuit composed of fan-in two AND/OR gates only. (In more detail, it is enough to set $d = \log N$ and replace each majority gate by a $O(\log N)$-depth fan-in-two AND/OR Boolean circuit.) We sketch a simpler construction in Appendix A that matches the parameters obtained in [BW05] in the case of the universal monotone threshold function.

Another consequence of our lower bound as stated in Theorem 1 is a significant strengthening of the Ajtai–Gurevich lower bound discussed earlier. We recall their result in more detail:

Theorem (Ajtai–Gurevich [AG87])**.** *There exists an explicit sequence $f = \{f_n\}_{n\in\mathbb{N}}$ of monotone Boolean functions $f_n\colon \{0,1\}^n \to \{0,1\}$ such that:*

- *(i)* $f \in \mathsf{AC}^0$*;*
- *(ii)* $f \notin \mathsf{monAC}^0$*: For any fixed constant d, any monotone depth-d AND/OR circuit computing f_n must have size at least $S_d(n)$, for some function $S_d(n) = n^{\omega(1)}$.*

Regarding part *(ii)* above, it is not immediately clear what is the best (largest) function $S_d(n)$ that can be extracted from the Ajtai–Gurevich proof. However, f_n is easily seen to be computed by a monotone depth-2 circuit (a monotone DNF) of size $n^{\log n}$, so $S_d(n) \le n^{\log n}$ for all $d \ge 2$.

As an easy corollary of Theorem 1, we strengthen the Ajtai–Gurevich circuit lower bound (for a different monotone function in AC^0) in two ways: by giving a lower bound against monotone *majority circuits* of constant depth (rather than monotone circuits of AND/OR gates only), and by achieving an *exponential* size lower bound for any fixed depth (rather than a bound which is at most $n^{\log n}$). Our theorem is the following:

Theorem 3. *There exists an explicit sequence $g = \{g_n\}_{n\in\mathbb{N}}$ of monotone Boolean functions, where $g_n\colon \{0,1\}^{n\log n} \to \{0,1\}$, such that:*

- *(i)* $g \in \mathsf{AC}^0$ *(in fact each g_n is computed by a* poly(n)*-size, depth-*3 AND/OR/NOT *circuit);*
- *(ii) For any constant $d \ge 2$, any monotone depth-d majority circuit for g_n must have size $2^{\Omega(n^{1/d})}$.*

It is interesting to observe that our proof of Theorem 3 uses very different arguments from those of Ajtai and Gurevich. The heart of their proof is a "switching lemma" for monotone functions on hypergrids (see the excellent exposition of their proof given in [BST13]), whereas our approach does not use switching lemmas at all.

1.2 Related Work and Our Techniques.

In addition to papers discussed above, the works of Yao [Yao89] and Håstad and Goldmann [HG91] are relevant in the context of our lower bound result. Let Sipser_{d+1} denote the read-once monotone n-variable formula of depth $d+1$ that has alternating layers of AND and OR gates (see [HG91] for a detailed description of this function). Strengthening the earlier result of Yao [Yao89], Håstad and Goldmann [HG91] showed that a depth-d circuit of *weighted* monotone threshold gates computing Sipser_{d+1} must have size $2^{\Omega(n^{1/2d})}$. In contrast, our Theorem 1 only establishes a lower bound against constant-depth monotone circuits of *unweighted* threshold gates, but — crucially — we establish the lower bound for a much "simpler" monotone function, $U_{d,N}$, that is computed by a *single* weighted monotone threshold gate. Indeed, the main challenge of our work is to push through a lower bound for such a heavily constrained target function.

At the heart of our lower bound proof is a sequence of carefully constructed pairs of probability distributions $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ over $\{0,1\}^{(\ell+1)\times N_\ell}$ for $\ell = 1, \ldots, d$ (i.e. over possible inputs to $U_{\ell+1,N_\ell}$ for some N_ℓ to be specified later). The first distribution $\mathcal{YES}_\ell$ in the pair is supported on strings x that have $U_{\ell+1,N_\ell}(x) = 1$, while $\mathcal{NO}_\ell$ is supported on strings with $U_{\ell+1,N_\ell}(x) = 0$. The key property of these pairs of distributions, which yields our lower bound, is that considered together, each pair of $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ is "hard" for "small" monotone majority circuits of depth ℓ in a suitable sense. In a bit more detail, our requirement is roughly that for any such circuit F, we have

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_\ell}\left[F(\boldsymbol{x}) = 1\right] + \mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}_\ell}\left[F(\boldsymbol{y}) = 0\right] \leq 1 + \tau_\ell, \tag{1}$$

for a suitable value $0 < \tau_\ell \ll 1$. At a high level, we establish (1) above through a careful inductive argument on ℓ. (We note that the preceding sketch is something of an oversimplification; actually, in order for the inductive hypothesis to be "strong enough to prove itself," we require an analogue of (1) both for the pair $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ and for another pair of distributions $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$, and the inductive argument establishing the case $\ell = j+1$ from the case $\ell = j$ requires careful analysis of yet a third carefully constructed pair $(\mathcal{YES}^*_\ell, \mathcal{NO}^*_\ell)$ of distributions. See Section 2 for full details of the argument.)[3]

Notation and Organization. Recall that a *restriction* ρ of a function f is an assignment fixing some of the input variables of f. We write "$f \restriction \rho$" to denote f restricted by ρ, a function over the rest of variables. We use boldface lower-case letters $\boldsymbol{x}, \boldsymbol{y}$, etc. to denote string-valued random variables and boldface capital letters $\mathbf{X}, \mathbf{Y}$, etc. to denote real-valued random variables.

The rest of the paper is organized as follows. We prove Theorems 1 and 2 in Sections 2 and 3, respectively. We then use Theorem 1 to prove Theorem 3 in Section 4.

2 The Lower Bound: Proof of Theorem 1.

We prove Theorem 1 in this section. Throughout the section we use d, n and N to denote the three positive integers in the statement of Theorem 1 with $n \geq 2^{60d}$ and $N \geq (2^{13}n)^d$.

This section is organized as follows. In Sections 2.1 and 2.2, we define inductively two pairs $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ and $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ of distributions over strings $\boldsymbol{x} \in \{0,1\}^{(\ell+1)\times N_\ell}$ for ℓ from 1 to d, where N_ℓ is specified later and satisfies $N_1 < \cdots < N_d \leq N$. An important property of these distributions is that every $\boldsymbol{x}$ drawn from $\mathcal{YES}_\ell$, $\mathcal{NO}_\ell$, $\mathcal{YES}'_\ell$, $\mathcal{NO}'_\ell$ has $\mathsf{SUM}(\boldsymbol{x})$ equal to

$$2^{N_\ell}, \quad 2^{N_\ell} - 1, \quad 2^{N_\ell} - 1 \quad \text{and} \quad 2^{N_\ell} - (\ell+1),$$

respectively. From the definition of $(\mathcal{YES}_1, \mathcal{NO}_1)$ and $(\mathcal{YES}'_1, \mathcal{NO}'_1)$, it is not too difficult to show that both pairs are *very hard* for monotone depth-1 majority circuits (Lemma 2.1), i.e. no majority gate with small weights can output 1 on strings drawn from $\mathcal{YES}_1$ with probability p_1 and at the same time output 0 on strings drawn from $\mathcal{NO}_1$ with probability p_2 if $p_1 + p_2$ is slightly larger than 1 (and the same holds for $\mathcal{YES}'_1$ and $\mathcal{NO}'_1$).

[3] Notice that the argument we just sketched implies that $U_{d+1,N}$ is hard against depth-d circuits. A more careful analysis at the end of the argument using the distributions $(\mathcal{YES}^*_d, \mathcal{NO}^*_d)$ allows us to obtain the same lower bound for $U_{d,N}$, as stated in Theorem 1.

Then we prove our main technical lemma (Lemma 2.7) in Section 2.3, which shows by induction that both pairs $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ and $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ are hard in the same sense for "small" depth-ℓ majority circuits over $\{0,1\}^{(\ell+1)\times N_\ell}$ for every $\ell \in [d]$, with $(\mathcal{YES}_1, \mathcal{NO}_1)$ and $(\mathcal{YES}'_1, \mathcal{NO}'_1)$ serving as the base case. Theorem 1 for $U_{d+1,N}$ (instead of $U_{d,N}$ as stated) follows directely from $N_d \le N$ and the property that strings $\boldsymbol{x}$ drawn from $\mathcal{YES}_d$ and $\mathcal{NO}_d$ have $\mathsf{SUM}(\boldsymbol{x})$ equal to 2^{N_d} and $2^{N_d}-1$, respectively. (Note that, although the second pair $(\mathcal{YES}'_d, \mathcal{NO}'_d)$ is not needed in the proof of Theorem 1 once Lemma 2.7 has been established, the intermediate pairs $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ play a crucial role in the inductive definition of these distributions and the proof of Lemma 2.7.)

In order to extend the result to $U_{d,N}$ (as stated in Theorem 1), we rely on another auxiliary pair of distributions $(\mathcal{YES}^*_d, \mathcal{NO}^*_d)$ constructed during the proof, which is described in more detail in Section 2.2. We finally use Lemma 2.7 to prove Theorem 1 in Section 2.4.

2.1 The Initial Two Pairs of Distributions.

Let d, n, N be positive integers in the statement of Theorem 1. Let $\varepsilon \stackrel{\text{def}}{=} 2^{-12d}$ and $N_1 \stackrel{\text{def}}{=} n \cdot (1/\varepsilon)$. Given a string $z \in \{0,1\}^{2\times N_1}$, the *$j$-th column of z* corresponds to a pair of positions $(1,j)$ and $(2,j)$, where $j \in [N_1]$.

We now define two pairs of distributions $(\mathcal{YES}_1, \mathcal{NO}_1)$ and $(\mathcal{YES}'_1, \mathcal{NO}'_1)$ over $\{0,1\}^{2\times N_1}$ and show that they are hard for monotone depth-1 majority circuits of not-too-large size. We define the distributions via the following sampling processes.

- A string $\boldsymbol{x} \sim \mathcal{YES}_1$ is generated as follows. Let $\boldsymbol{R} \sim [N_1]$ be uniformly random. We set both bits in the $\boldsymbol{R}$-th column of $\boldsymbol{x}$ to 1. For every $j > \boldsymbol{R}$, we set both bits in the j-th column of $\boldsymbol{x}$ to 0. For every $j < \boldsymbol{R}$, we set the j-th column of $\boldsymbol{x}$ to $(1,0)$ or $(0,1)$ independently and with equal probability. For example, writing an $x \in \mathrm{supp}(\mathcal{YES}_1)$ as a matrix, it would look like

$$x = \left[\begin{array}{cccccc|c|cccccc} 1 & 0 & 0 & 1 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & 0 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] \quad \text{and we have} \quad \mathsf{SUM}(x) = 2^{N_1}.$$

- A string $\boldsymbol{y} \sim \mathcal{NO}_1$ is generated by setting its j-th column to $(1,0)$ or $(0,1)$ independently and with equal probability for each $j \in [\boldsymbol{R}]$. So a string $y \in \mathrm{supp}(\mathcal{NO}_1)$ would look like

$$y = \begin{array}{cccccccccccc} 0 & 1 & 0 & 0 & \cdots & 1 & 1 & 0 & 1 & \cdots & 1 & 0 \\ 1 & 0 & 1 & 1 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 & 1 \end{array} \quad \text{and we have} \quad \mathsf{SUM}(y) = 2^{N_1} - 1.$$

- $\mathcal{YES}'_1$ is the same as $\mathcal{NO}_1$. In particular, each $x \in \mathrm{supp}(\mathcal{YES}'_1)$ has $\mathsf{SUM}(x) = 2^{N_1} - 1$.

- Finally, a string $\boldsymbol{y} \sim \mathcal{NO}'_1$ is obtained as follows. First, sample a random $\boldsymbol{x} \sim \mathcal{YES}_1$. Then let $\boldsymbol{y}$ be the string obtained by negating each bit of $\boldsymbol{x}$. So a string $y \in \mathrm{supp}(\mathcal{NO}'_1)$ looks like

$$y = \left[\begin{array}{cccccc|c|cccccc} 1 & 1 & 0 & 1 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 1 & 0 & \cdots & 1 & 0 & 1 & 1 & \cdots & 1 & 1 \end{array}\right] \quad \text{and} \quad \mathsf{SUM}(y) = 2^{N_1} - 2.$$

Recall a monotone depth-1 majority circuit of size s is simply a monotone weighted majority gate with total weight at most s. We show below that both pairs $(\mathcal{YES}_1, \mathcal{NO}_1)$ and $(\mathcal{YES}'_1, \mathcal{NO}'_1)$ defined above are hard for a monotone depth-1 circuit (to be correct on both $\mathcal{YES}_1$ and $\mathcal{NO}_1$, or on both $\mathcal{YES}'_1$ and $\mathcal{NO}'_1$, with nontrivial probability) unless the total weight s is large.

Lemma 2.1. *For any depth-1 monotone majority circuit F over $\{0,1\}^{2\times N_1}$ of size at most 2^{n-1},*

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_1}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}_1}\left[F(\boldsymbol{y})=0\right]\leq 1+\varepsilon, \quad \textit{and} \tag{2}$$

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}'_1}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}'_1}\left[F(\boldsymbol{y})=0\right]\leq 1+\varepsilon. \tag{3}$$

Proof. We present the proof of the first inequality on $(\mathcal{YES}_1,\mathcal{NO}_1)$. An entirely similar argument establishes the bound for $(\mathcal{YES}'_1,\mathcal{NO}'_1)$.

Consider an auxiliary distribution $\mathcal{D}$ (essentially a coupling of $\mathcal{YES}_1$ and $\mathcal{NO}_1$) supported over $\{0,1\}^{2\times N_1}\times\{0,1\}^{2\times N_1}\times[N_1]$, and defined in the following way. A draw $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}$ is obtained by selecting a uniformly random $\boldsymbol{R}\sim[N_1]$, a string $\boldsymbol{y}\sim\mathcal{NO}_1$, and by letting $\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{y},\boldsymbol{R})\in\{0,1\}^{2\times N_1}$ be the string obtained by replacing the $\boldsymbol{R}$-th column of $\boldsymbol{y}$ with $(1,1)$, and by setting the j-th column of $\boldsymbol{y}$ to $(0,0)$ for every $j>\boldsymbol{R}$. Observe that the marginal distributions $\mathcal{D}_{\boldsymbol{x}}$ and $\mathcal{D}_{\boldsymbol{y}}$ are identical to $\mathcal{YES}_1$ and $\mathcal{NO}_1$, respectively. Consequently,

$$\begin{aligned}\text{LHS of Equation (2)} &= \mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[F(\boldsymbol{y})=0\right]\\ &= \mathbf{Pr}\left[F(\boldsymbol{x})=1 \text{ or } F(\boldsymbol{y})=0\right]+\mathbf{Pr}\left[F(\boldsymbol{x})=1 \text{ and } F(\boldsymbol{y})=0\right]\\ &\leq 1+\mathbf{Pr}\left[F(\boldsymbol{x})=1 \text{ and } F(\boldsymbol{y})=0\right].\end{aligned}$$

Hence to prove the lemma, it is enough to show that

$$q\stackrel{\text{def}}{=}\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[F(\boldsymbol{x})=1 \text{ and } F(\boldsymbol{y})=0\right]\leq\varepsilon. \tag{4}$$

For every $r\in[N_1]$, let $\mathbf{Y}_r$ be an indicator random variable defined on $\mathcal{D}$ that is 1 whenever

$$w_r(\boldsymbol{y}) \;>\; \sum_{\ell>r} w_\ell(\boldsymbol{y}),$$

where $w_j(y)\stackrel{\text{def}}{=}w_{1,r}\cdot y_{1,r}+w_{2,r}\cdot y_{2,r}$, and $w_{i,j}$ is the weight corresponding to the input variable of F at position (i,j). Informally, $\mathbf{Y}_r=1$ if and only if the weight of $\boldsymbol{y}$ with respect to F at the r-th column is strictly larger than the sum of the weights collected from all succeeding columns.

We will employ the following claim to establish Equation (4).

Claim 2.2. *For every $j\in[N_1]$, we have*

$$q_j\stackrel{\text{def}}{=}\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[F(\boldsymbol{x})=1 \text{ and } F(\boldsymbol{y})=0\,\middle|\,\boldsymbol{R}=j\right]\leq\mathbf{Pr}_{\mathcal{D}}\left[\mathbf{Y}_j=1\right].$$

Proof. We consider first the case where $j=1$. The conditions of $F(\boldsymbol{x})=1$ and $\boldsymbol{R}=1$ imply that $w_{1,1}+w_{2,1}\geq t$, where t is the threshold of F. Furthermore, because $F(\boldsymbol{y})=0$ it must be the case that $\sum_{r=1}^{N_1}w_r(\boldsymbol{y})<t$. These inequalities give us

$$w_{1,1}+w_{2,1}-w_1(\boldsymbol{y})>\sum_{r>1}w_r(\boldsymbol{y}). \tag{5}$$

Let $\widetilde{\boldsymbol{y}}$ be the string obtained from $\boldsymbol{y}$ by flipping the two bits in the first column of $\boldsymbol{y}$. Equation (5)

is then equivalent to $w_1(\widetilde{\boldsymbol{y}}) > \sum_{r>1} w_r(\widetilde{\boldsymbol{y}})$. Therefore,

$$\begin{aligned} q_1 &\leq \mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[w_1(\widetilde{\boldsymbol{y}}) > \sum_{r>1} w_r(\widetilde{\boldsymbol{y}}) \,\middle|\, \boldsymbol{R}=1\right] \\ &= \mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[w_1(\boldsymbol{y}) > \sum_{r>1} w_r(\boldsymbol{y}) \,\middle|\, \boldsymbol{R}=1\right] \\ &= \mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{R})\sim\mathcal{D}}\left[\mathbf{Y}_1 = 1\right], \end{aligned}$$

where the last two equations use the independence of $\boldsymbol{y}$ and $\boldsymbol{R}$ as well as the fact that $\widetilde{\boldsymbol{y}}$ and $\boldsymbol{y}$ are identically distributed.

For $j > 1$ the result can be proved similarly by writing q_j as a conditional expectation over the outcome of the first $j-1$ columns of $\boldsymbol{y}$, then adapting the argument above in the natural way. □

Claim 2.2 and the definitions of probabilities q and q_j imply that

$$N_1 \cdot q \;=\; \sum_{j=1}^{N_1} q_j \;\leq\; \sum_{j=1}^{N_1} \mathbf{Pr}_{\mathcal{D}}\left[\mathbf{Y}_j = 1\right] \;=\; \mathbf{E}_{\mathcal{D}}\left[\sum_{j=1}^{N_1} \mathbf{Y}_j\right].$$

In particular, there is a string $y^* \in \mathrm{supp}(\mathcal{NO}_1)$ and a set $S \subseteq [N_1]$ with $|S| \geq N_1 \cdot q$ such that

$$w_r(y^*) > \sum_{\ell > r} w_\ell(y^*), \tag{6}$$

for each $r \in S$. Recall that the weight associated to each variable in F is a non-negative integer, and that the total weight of F is at least $\sum_{r\geq 1} w_r(y^*)$. It follows directly from (6) that F must have total weight at least $2^{|S|-1}$. However, by assumption F has total weight at most 2^{n-1}. Altogether, we get from these inequalities and $N_1 = n \cdot (1/\varepsilon)$ that $q \leq \varepsilon$, which completes the proof. □

2.2 A Sequence of Pairs of Pairs of Distributions.

Next, suppose that we have defined pairs of distributions $(\mathcal{YES}_{\ell-1}, \mathcal{NO}_{\ell-1})$ and $(\mathcal{YES}'_{\ell-1}, \mathcal{NO}'_{\ell-1})$ over $\{0,1\}^{\ell \times N_{\ell-1}}$ for some $2 \leq \ell \leq d$, where a string $\boldsymbol{x}$ drawn from $\mathcal{YES}_{\ell-1}$, $\mathcal{NO}_{\ell-1}$, $\mathcal{YES}'_{\ell-1}$ and $\mathcal{NO}'_{\ell-1}$ has $\mathsf{SUM}(\boldsymbol{x})$ equal to

$$2^{N_{\ell-1}}, \quad 2^{N_{\ell-1}} - 1, \quad 2^{N_{\ell-1}} - 1 \quad \text{and} \quad 2^{N_{\ell-1}} - ((\ell-1)+1), \tag{7}$$

respectively. (Note that the pairs $(\mathcal{YES}_1, \mathcal{NO}_1)$ and $(\mathcal{YES}'_1, \mathcal{NO}'_1)$ have this property.) Our aim is to inductively define $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ and $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ over $\{0,1\}^{(\ell+1)\times N_\ell}$, where

$$N_\ell \stackrel{\text{def}}{=} n \cdot N_{\ell-1} + 1 \leq 2^\ell \cdot n^{\ell-1} \cdot N_1 = (2n)^\ell \cdot 2^{12^d} \leq (2^{13}n)^d \leq N, \quad \text{for } \ell \in \{2, \ldots, d\},$$

and a string $\boldsymbol{x}$ drawn from $\mathcal{YES}_\ell$, $\mathcal{NO}_\ell$, $\mathcal{YES}'_\ell$ and $\mathcal{NO}'_\ell$ has $\mathsf{SUM}(\boldsymbol{x})$ equal to

$$2^{N_\ell}, \quad 2^{N_\ell} - 1, \quad 2^{N_\ell} - 1 \quad \text{and} \quad 2^{N_\ell} - (\ell+1), \tag{8}$$

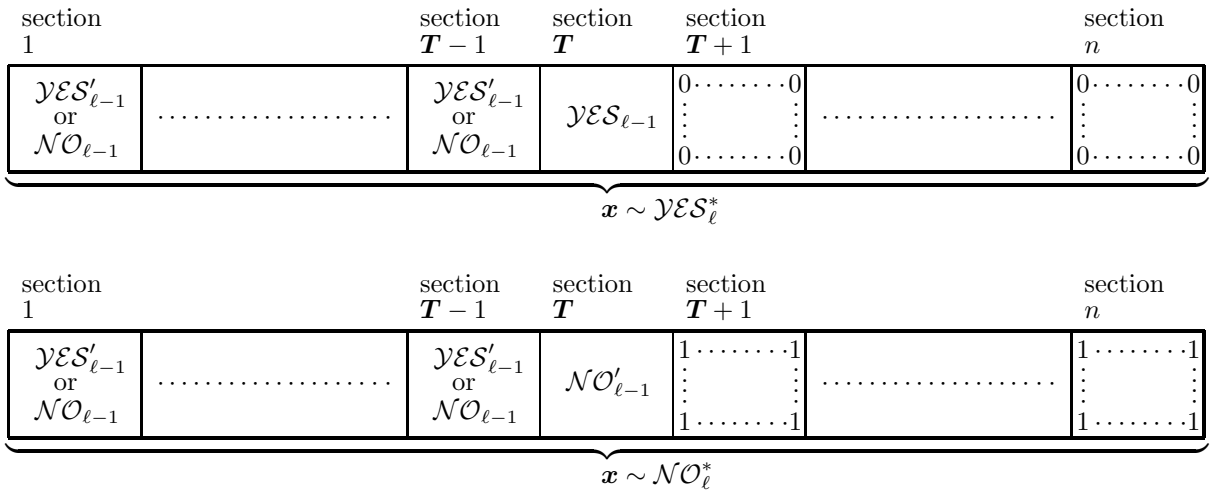

Figure 1: Illustrations of how the $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ distributions are defined from the $\mathcal{YES}'_{\ell-1}, \mathcal{NO}'_{\ell-1}, \mathcal{YES}_{\ell-1}$ and $\mathcal{NO}_{\ell-1}$ distributions.

respectively. To this end we start by defining a pair of distributions $(\mathcal{YES}^*_\ell, \mathcal{NO}^*_\ell)$ over $\{0,1\}^{\ell \times N^*_\ell}$ (note that the number of rows for these distributions, ℓ, is exactly the same as for the distributions $\mathcal{YES}_{\ell-1}, \mathcal{NO}_{\ell-1}$ and $\mathcal{YES}'_{\ell-1}, \mathcal{NO}'_{\ell-1}$), with

$$N^*_\ell \stackrel{\text{def}}{=} n \cdot N_{\ell-1} = N_\ell - 1.$$

To define $(\mathcal{YES}^*_\ell, \mathcal{NO}^*_\ell)$, we partition the N^*_ℓ columns into n *sections*, each with $N_{\ell-1}$ columns (and ℓ rows). (So the first section consists of all $x_{i,j}$ with $j \in [N_{\ell-1}]$, the second section consists of all $x_{i,j}$ with $j \in [N_{\ell-1}+1, 2N_{\ell-1}]$, and so forth.) A draw of a string from $\mathcal{YES}^*_\ell$ is obtained as follows: first we draw an integer $\boldsymbol{T}$ uniformly from $[n]$, and then

- (a) For each $i < \boldsymbol{T}$, we independently set the i-th section to be a string drawn from $\mathcal{NO}_{\ell-1}$ with probability $1/2$ or a string drawn from $\mathcal{YES}'_{\ell-1}$ with probability $1/2$.
- (b) For each $i > \boldsymbol{T}$, we set the i-th section to be all 0.
- (c) For the $\boldsymbol{T}$-th section, we set it to be a string drawn from $\mathcal{YES}_{\ell-1}$.

See Figure 1 for an illustration. A draw of a string from $\mathcal{NO}^*_\ell$ is obtained in a similar fashion. First we draw $\boldsymbol{T}$ from $[n]$ uniformly at random, and then

- (a') For each $i < \boldsymbol{T}$, we independently set the i-th section to be a string drawn from $\mathcal{NO}_{\ell-1}$ with probability $1/2$ or a string drawn from $\mathcal{YES}'_{\ell-1}$ with probability $1/2$. (Note that this is the same as step (a) above in the definition of $\mathcal{YES}^*_\ell$.)
- (b') For each $i > \boldsymbol{T}$, we set the i-th section to be all 1 (this is different from (b) above).
- (c') For the $\boldsymbol{T}$-th section, we set it to be a string drawn from $\mathcal{NO}'_{\ell-1}$ (this is different from (c)).

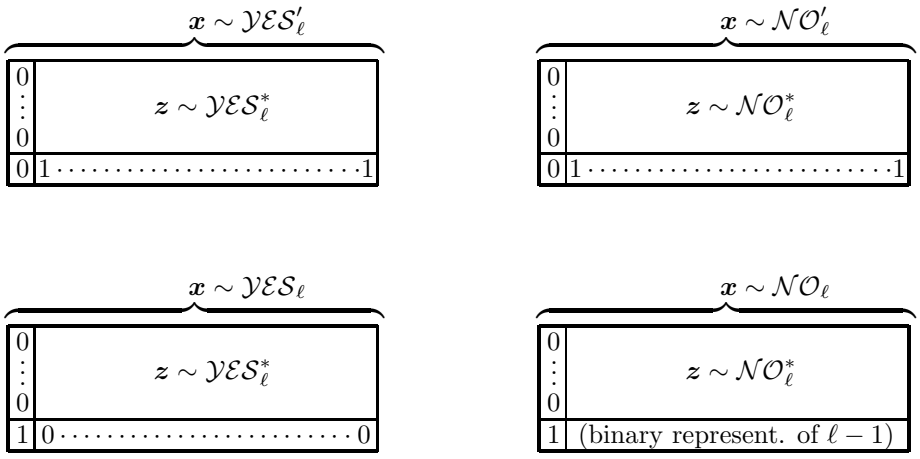

Figure 2: Illustrations of how the $\mathcal{YES}'_\ell, \mathcal{NO}'_\ell, \mathcal{YES}_\ell$ and $\mathcal{NO}_\ell$ distributions are defined from the $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ distributions.

Again see Figure 1 for an illustration. Given (7), we see that a string $\boldsymbol{x}$ drawn from $\mathcal{YES}^*_\ell$ (or from $\mathcal{NO}^*_\ell$) has $\mathsf{SUM}(\boldsymbol{x})$ equal to $2^{N^*_\ell}$ (respectively, equal to $2^{N^*_\ell} - \ell$).

With the definitions of $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ in hand, we now use them to define $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ and $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ so that every string $\boldsymbol{x}$ drawn from these distributions should have $\mathsf{SUM}(\boldsymbol{x})$ equal to the values given in (8). Recall that $N_\ell = N^*_\ell + 1$.

A string $\boldsymbol{x} = (x_{i,j}) \in \{0,1\}^{(\ell+1)\times N_\ell}$ drawn from $\mathcal{YES}'_\ell$ is obtained as follows. First we draw a string $\boldsymbol{z}$ from $\mathcal{YES}^*_\ell$ and put it in columns $\{2, \ldots, N_\ell\}$ and rows $\{1, \ldots, \ell\}$ of $\boldsymbol{x}$, i.e., $x_{i,j} = z_{i,j-1}$ for all $i \in [\ell]$ and $j \in \{2, \ldots, N_\ell\}$. For the remaining positions (in the first column and the last row), we set $x_{i,1} = 0$ for all $i \in [\ell + 1]$ and $x_{\ell+1,j} = 1$ for all $j > 1$. The other distribution $\mathcal{NO}'_\ell$ is defined similarly, except that we draw the string $\boldsymbol{z}$ from $\mathcal{NO}^*_\ell$ instead of from $\mathcal{YES}^*_\ell$. The definition of $\mathcal{YES}'_\ell$ and $\mathcal{NO}'_\ell$ is illustrated in Figure 2.

For the other pair $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$, a string $\boldsymbol{x}$ drawn from $\mathcal{YES}_\ell$ is obtained as follows. As before, we first draw a string $\boldsymbol{z}$ from $\mathcal{YES}^*_\ell$ and put it in columns $\{2, \ldots, N_\ell\}$ and rows $\{1, \ldots, \ell\}$ of $\boldsymbol{x}$. Then we set $x_{\ell+1,1} = 1$ and all other variables on the first row and last column of $\boldsymbol{x}$ to be 0. For the other distribution $\mathcal{NO}_\ell$, we similarly draw $\boldsymbol{z}$ from $\mathcal{NO}^*_\ell$ and put it in columns $\{2, \ldots, N_\ell\}$ and rows $\{1, \ldots, \ell\}$ of $\boldsymbol{x}$. We set $x_{\ell+1,1} = 1$ and all other variables on the first column to be 0. We set the last row, i.e., $x_{\ell+1,j}$ with $j \in \{2, \ldots, N_\ell\}$, to be the binary representation of $\ell - 1$. (This is well defined since $N^*_\ell \geq n \geq 2^{60d} \gg \log d \geq \log \ell$.) As before, see Figure 2 for an illustration of the definition of $\mathcal{YES}'_\ell$ and $\mathcal{NO}'_\ell$.

We record the following two useful facts about N_ℓ and the distributions:

Fact 2.3. $N_d \leq N$.

Fact 2.4. *For each $\ell \in [d]$, a string $\boldsymbol{x}$ drawn from $\mathcal{YES}_\ell$, $\mathcal{NO}_\ell$, $\mathcal{YES}'_\ell$, $\mathcal{NO}'_\ell$ has $\mathsf{SUM}(\boldsymbol{x})$ equal to*

$$2^{N_\ell}, \quad 2^{N_\ell} - 1, \quad 2^{N_\ell} - 1 \quad \text{and} \quad 2^{N_\ell} - (\ell + 1).$$

An important property of the $(\mathcal{YES}_\ell, \mathcal{NO}_\ell)$ pair and of the $(\mathcal{YES}'_\ell, \mathcal{NO}'_\ell)$ pair — which in fact motivated the above definitions of these distributions in terms of $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ — is that they are *at least as hard* to distinguish as $(\mathcal{YES}^*_\ell, \mathcal{NO}^*_\ell)$ for monotone majority circuits.

This is made formal in the following two lemmas.

Lemma 2.5. *Given any monotone majority circuit F over $\{0,1\}^{(\ell+1)\times N_\ell}$, there is a monotone majority circuit F^* over $\{0,1\}^{\ell\times N^*_\ell}$ of the same size and depth as F such that*

$$\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}'_\ell}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}'_\ell}\left[F(\boldsymbol{y})=0\right]=\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right].$$

Proof. Given F, we hard-wire the variables in the first column to be 0 and the rest of the variables in the last row to be 1. Let F^* denote the new monotone majority circuit obtained from F of the same size and depth. The definition of $\mathcal{YES}'_\ell$ and $\mathcal{NO}'_\ell$ from $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ implies that

$$\begin{aligned}\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}'_\ell}\left[F(\boldsymbol{x})=1\right]&=\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]\quad\text{and}\\ \mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}'_\ell}\left[F(\boldsymbol{y})=0\right]&=\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right].\end{aligned}$$

The lemma then follows. □

Lemma 2.6. *Given any monotone majority circuit F over $\{0,1\}^{(\ell+1)\times N_\ell}$, there is a monotone majority circuit F^* over $\{0,1\}^{\ell\times N^*_\ell}$ of the same size and depth as F such that*

$$\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}_\ell}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}_\ell}\left[F(\boldsymbol{y})=0\right]\le\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right].$$

Proof. Given F, we hard-wire $x_{\ell+1,1}$ to be 1 and the rest of the variables in the first column and the last row to be 0. Let F^* denote the resulting monotone majority circuit obtained from F of the same size and depth. The definition of $\mathcal{YES}_\ell$ and $\mathcal{NO}_\ell$ from $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ implies that

$$\begin{aligned}\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}_\ell}\left[F(\boldsymbol{x})=1\right]&=\mathbf{Pr}_{\boldsymbol{x}\in\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]\quad\text{and}\\ \mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}_\ell}\left[F(\boldsymbol{y})=0\right]&\le\mathbf{Pr}_{\boldsymbol{y}\in\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right],\end{aligned}$$

where the inequality follows from the monotonicity of F. The lemma then follows. □

2.3 The Key Induction Lemma.

Given distributions defined in Sections 2.1 and 2.2, we prove the following key technical lemma.

Recall that $\varepsilon = 2^{-12d}$. Below we let $M = 2^{\varepsilon^5 n}$.

Lemma 2.7. *Let $\ell \in \{2,\dots,d\}$. Suppose that any depth-$(\ell-1)$ monotone majority circuit F over $\{0,1\}^{\ell\times N_{\ell-1}}$ of size at most M satisfies*

$$\begin{aligned}&\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_{\ell-1}}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}_{\ell-1}}\left[F(\boldsymbol{y})=0\right]\le 1+7^{\ell-2}\varepsilon\quad\textit{and}\\ &\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}'_{\ell-1}}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}'_{\ell-1}}\left[F(\boldsymbol{y})=0\right]\le 1+7^{\ell-2}\varepsilon.\end{aligned}\tag{9}$$

Then any depth-ℓ monotone majority circuit F^* over $\{0,1\}^{\ell\times N^*_\ell}$ of size at most M satisfies

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right]\le 1+7^{\ell-1}\varepsilon.$$

Proof. Recall that strings drawn from $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ consist of n sections. For convenience, we refer to strings in $\{0,1\}^{\ell \times N_{\ell-1}}$ as *section strings*.

We begin by defining some useful distributions $\mathcal{D}_1, \ldots, \mathcal{D}_n$ over concatenations of section strings where $\mathcal{D}_t$ is supported on concatenations of $t-1$ section strings. First, let $\mathcal{D}$ denote the following distribution over section strings: $\boldsymbol{x} \sim \mathcal{D}$ is drawn from $\mathcal{NO}_{\ell-1}$ with probability $1/2$ and is drawn from $\mathcal{YES}'_{\ell-1}$ with probability $1/2$. For each $t \in [n]$, we use $\mathcal{D}_t$ to denote the distribution of the concatenation of $t-1$ section strings, each drawn from $\mathcal{D}$ independently. (So $\mathcal{D}_t$ is a distribution over $\{0,1\}^{\ell \times (t-1) N_{\ell-1}}$.) Note that in the special case when $t = 1$, $\mathcal{D}_1$ is supported on the empty string only. Note also that for $t \in [n]$, $\mathcal{D}_t$ is generated precisely according to (a) or (a') from Section 2.2 (recall that (a) and (a') are the same).

As in the statement of Lemma 2.7, let F^* be a depth-ℓ monotone majority circuit on $\{0,1\}^{\ell \times N^*_\ell}$ of size at most M. We say a string $z \in \mathrm{supp}(\mathcal{D}_t)$ for some $t \in [n]$ is *good* with respect to F^* if

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{YES}_{\ell-1}}\left[F^*(z \circ \boldsymbol{x} \circ \mathbf{0}) = 1\right] + \mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{NO}'_{\ell-1}}\left[F^*(z \circ \boldsymbol{y} \circ \mathbf{1}) = 0\right] \geq 1 + 6\delta,$$

where we write $\mathbf{0}$ and $\mathbf{1}$ to denote the all-0 and all-1 strings in $\{0,1\}^{\ell \times (n-t) N_\ell}$, and $\delta \stackrel{\text{def}}{=} 7^{\ell-2}\varepsilon$.

Now we fix a $t \in [n]$ and fix a good string $z \in \mathrm{supp}(\mathcal{D}_t)$. Let ρ_z be the restriction that fixes the first $t-1$ sections of variables of F^* to be z and leaves the remaining $n-(t-1)$ sections unfixed. As z is good, we have that $F^* \restriction \rho_z$ is nontrivial (i.e., $F^* \restriction \rho_z \not\equiv 0$ or 1). We write $H_1, \ldots, H_m$ (with multiplicities) to denote the set of all depth-$(\ell-1)$ sub-circuits rooted at children of the output gate of F^* such that $H_i \restriction \rho_z$ is nontrivial. In other words, we assume that the same sub-circuit may appear multiple times in this list if the output majority gate in F^* contains multiple wires to it. Since the size of (F^*) is at most M, the fan-in of the output majority gate of F^* is at most M, and consequently $m \leq M$. Since $F^* \restriction \rho$ is nontrivial there is a positive integer $h \in [M]$ such that $F^* \restriction \rho_z$ outputs 1 if and only if at least h many of $H_1 \restriction \rho_z, \ldots, H_m \restriction \rho_z$ output 1. The following claim shows that with non-negligible probability, a random $\boldsymbol{x} \sim \mathcal{D}$ is such that "many" H_i's become trivial (i.e., compute a constant function) after a restriction by $\rho_{z \circ \boldsymbol{x}}$:

Claim 2.8. *Suppose that z is a good string in the support of $\mathcal{D}_t$. Then we have*

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{D}}\left[\left|\{i \in [m] : H_i \restriction \rho_{z \circ \boldsymbol{x}} \text{ is trivial }\}\right| \geq \delta^2 m/2\right] \geq \delta/4.$$

Proof. We consider two cases: $h \geq m/2$ or $h < m/2$. We focus on the latter below and the former case is symmetric. Assume that $h < m/2$. Since z is good, we have

$$\mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{NO}'_{\ell-1}}\left[F^*(z \circ \boldsymbol{y} \circ \mathbf{1}) = 0\right] \geq 1 + 6\delta - 1 = 6\delta.$$

However, if $y \in \mathrm{supp}(\mathcal{NO}'_{\ell-1})$ satisfies $F^*(z \circ y \circ \mathbf{1}) = 0$, then by $h < m/2$ it must be the case that at least $m/2$ of the H_i's have $H_i(z \circ y \circ \mathbf{1}) = 0$, and hence

$$\mathbf{E}_{\boldsymbol{y} \sim \mathcal{NO}'_{\ell-1}}\left[\text{number of } H_i\text{'s with } H_i(z \circ \boldsymbol{y} \circ \mathbf{1}) = 0\right] \geq 3\delta m. \tag{10}$$

Let I denote the set of $i \in [m]$ such that

$$\mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{NO}'_{\ell-1}}\left[H_i(z \circ \boldsymbol{y} \circ \mathbf{1}) = 0\right] \geq 2\delta. \tag{11}$$

Then we have from (10) that

$$|I| \cdot 1 + (m - |I|) \cdot 2\delta \geq 3\delta m,$$

which implies that $|I| \geq \delta m$.

We write ρ to denote the restriction over $\{0,1\}^{\ell \times N^*_\ell}$ that fixes the first $t-1$ sections of input variables to be z and the last $(n-t)$ sections of input variables to be all 1, and leaves only the variables in the t-th section unfixed. So each $H_i \restriction \rho$ is a depth-$(\ell-1)$ monotone majority circuit over $\{0,1\}^{\ell \times N_{\ell-1}}$ of size at most M. Then combining (11) and the assumption of the lemma, i.e., (9), applied to $H_i \restriction \rho$, we have that each $i \in I$ satisfies

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{YES}'_{\ell-1}} \left[H_i(z \circ \boldsymbol{x} \circ \mathbf{1}) = 1 \right] \leq 1 + \delta - 2\delta = 1 - \delta,$$

and thus,

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{YES}'_{\ell-1}} \left[H_i(z \circ \boldsymbol{x} \circ \mathbf{1}) = 0 \right] \geq \delta. \tag{12}$$

Note that if an $x \in \mathrm{supp}(\mathcal{YES}'_{\ell-1})$ satisfies $H_i(z \circ x \circ \mathbf{1}) = 0$, then we have $H_i \restriction \rho_{z \circ x} \equiv 0$ by the monotonicity of H_i. Let $\mathbf{X}$ be a random variable that denotes the number of H_i's that become trivial after $\rho_{z \circ \boldsymbol{x}}$, where $\boldsymbol{x} \sim \mathcal{YES}'_{\ell-1}$. So by (12) the expectation of $\mathbf{X}$ is at least $\delta|I|$. Let q denote the probability that $\mathbf{X} \geq \delta|I|/2$. The lower bound $\mathbf{E}[\mathbf{X}] \geq \delta|I|$ implies that

$$q \cdot |I| + (1-q) \cdot \delta|I|/2 \geq \delta|I|,$$

and thus $q \geq \delta/2$. Plugging in $|I| \geq \delta m$, we have that $\mathbf{X} \geq \delta^2 m/2$ with probability at least $\delta/2$.

Finally, taking into account that a draw of $\boldsymbol{x} \sim \mathcal{D}$ is drawn from $\mathcal{YES}'_{\ell-1}$ with probability $1/2$, we see that with probability at least $\delta/4$ over a draw of $\boldsymbol{x} \sim \mathcal{D}$, we have that at least $\delta^2 m/2$ many H_i's become trivial after $\rho_{z \circ \boldsymbol{x}}$. This finishes the proof of the claim. □

Claim 2.8 implies that if $\boldsymbol{z} \sim \mathcal{D}_t$ is good (with respect to F^*), then with probability at least $\delta/4$ over a random draw of $\boldsymbol{x} \sim \mathcal{D}$, the restriction $\rho_{\boldsymbol{z} \circ \boldsymbol{x}}$ trivializes at least $(\delta^2/2)$-fraction of the depth-$(\ell-1)$ sub-circuits of F that are *not* trivialized by $\rho_{\boldsymbol{z}}$. Intuitively, this is useful because it means that we have a good chance of getting a significant simplification of F (shrinking the fan-in of the top gate by a lot), and since F is of size at most M this cannot happen too many times. On the other hand, note that if $\boldsymbol{z}$ is not good, then by definition we have

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{YES}_{\ell-1}} \left[F^*(\boldsymbol{z} \circ \boldsymbol{x} \circ \mathbf{0}) = 1 \right] + \mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{NO}'_{\ell-1}} \left[F^*(\boldsymbol{z} \circ \boldsymbol{y} \circ \mathbf{1}) = 0 \right] < 1 + 6\delta,$$

which intuitively is also useful for our purpose of bounding

$$\mathbf{Pr}_{\boldsymbol{x} \sim \mathcal{YES}^*_\ell} \left[F^*(\boldsymbol{x}) = 1 \right] + \mathbf{Pr}_{\boldsymbol{y} \sim \mathcal{NO}^*_\ell} \left[F^*(\boldsymbol{y}) = 0 \right] \tag{13}$$

from above by $1 + 7\delta$.

To finish the proof of the lemma, we take the following alternative but equivalent view of (13). Let $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ be a sequence of random section strings, each drawn from $\mathcal{D}$ independently. By the

definition of $\mathcal{YES}^*_\ell$ and $\mathcal{NO}^*_\ell$ (recall Figure 1), we have that

$$(13) \times n \;=\; \mathbf{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_n}\left[\sum_{t=1}^{n} \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_{\ell-1}}\left[F^*(\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}\circ\boldsymbol{x}\circ\mathbf{0})=1\right] + \sum_{t=1}^{n}\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}'_{\ell-1}}\left[F^*(\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}\circ\boldsymbol{y}\circ\mathbf{1})=0\right]\right].$$

This can be viewed as the expectation of a random variable $\mathbf{\Gamma}$ generated as follows.

1. Start with $\mathbf{\Gamma}=0$.
2. For each "round" $t=1,\ldots,n$, independently draw $\boldsymbol{z}_t$ from $\mathcal{D}$ and add the following to $\mathbf{\Gamma}$:

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_{\ell-1}}\left[F^*(\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}\circ\boldsymbol{x}\circ\mathbf{0})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}'_{\ell-1}}\left[F^*(\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}\circ\boldsymbol{y}\circ\mathbf{1})=0\right].$$

So it suffices to show that $\mathbf{E}[\mathbf{\Gamma}]\le(1+7\delta)n$.

For each of the n rounds $t=1,\ldots,n$, exactly one of the following two possibilities must hold:

1. The current string $\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}$ is not good. In this case $\mathbf{\Gamma}$ goes up by at most $1+6\delta$ in the t-th round. Otherwise,

2. The current string $\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}$ is good. In this case $\mathbf{\Gamma}$ can go up by at most 2 in the t-th round, but by our previous analysis (specifically, Claim 2.8), the number of nontrivial depth-$(\ell-1)$ subcircuits of F^* (with multiplicities) rooted at children of the output gate of F^* drops by a factor of $(1-\delta^2/2)$ with probability at least $\delta/4$ when the draw of $\boldsymbol{z}_t$ in the t-th round extends the restriction to $\rho_{\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_t}$. Note that F^* has size $M\le 2^{\varepsilon^5 n}$ so it can survive at most $2\delta^3 n$ many such drops before F^* becomes trivial; to see this, observe that

$$(1-\delta^2/2)^{2\delta^3 n}\le\exp\left(-(\delta^2/2)\cdot(2\delta^3 n)\right)=\exp\left(-\delta^5 n\right)<2^{-\varepsilon^5 n}. \tag{14}$$

   Note further that once F^* becomes trivial, $\mathbf{\Gamma}$ goes up by 1 in every subsequent round.

We use $\mathbf{S}$, a random variable, to denote the total number of rounds $t\in[n]$ such that the current string $\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}$ is good (note that once F^* becomes trivial the current string cannot be good). We claim that $\mathbf{S}\le 32\delta^2 n$ with high probability.

Claim 2.9. *We have $\mathbf{S}\le 32\delta^2 n$ with probability at least $1-\exp(-n\delta^4/2)$.*

Proof. We say that round t is good if the current string $\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}$ is good. We say that F^* is *hit* in the t-th round, if $\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}$ is good and the number of depth-$(\ell-1)$ subcircuitscuits of F^* (with multiplicities) that are trivial under the restriction $\rho_{\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}}$ drops by a factor of at least $(1-\delta^2/2)$ under the restriction $\rho_{\boldsymbol{z}_1\circ\cdots\circ\boldsymbol{z}_{t-1}\circ\boldsymbol{z}_t}$. Then we can write $\mathbf{Pr}[\mathbf{S}\ge 32\delta^2 n]$ as

$$\mathbf{Pr}\left[\mathbf{S}\ge 32\delta^2 n \;\&\; F^* \text{ is hit} > 2\delta^3 n \text{ many times during the first } 32\delta^2 n \text{ of the good rounds}\right]$$
$$+\,\mathbf{Pr}\left[\mathbf{S}\ge 32\delta^2 n \;\&\; F^* \text{ is hit} \le 2\delta^3 n \text{ many times during the first } 32\delta^2 n \text{ of the good rounds}\right].$$

The first of these probabilities is zero because of (14), i.e. if F^* is hit $2\delta^3 n$ times then it is trivialized so no subsequent rounds can be good and thus F^* cannot be hit again.

We focus on upper bounding the second probability. For each i from 1 to $32\delta^2 n$ we define the following random variable $\mathbf{Y}_i$ where

$$\mathbf{Y}_i = \begin{cases} 1 & \text{if } F^* \text{ is hit in the } i\text{-th good round or there are fewer than } i \text{ good rounds} \\ 0 & \text{otherwise (there are at least } i \text{ good rounds and } F^* \text{ is not hit in the } i\text{th good round).} \end{cases}$$

The second probability we are interested in is at most $\mathbf{Pr}[\sum_i \mathbf{Y}_i \le 2\delta^3 n]$. By Claim 2.8, we have

$$\mathbf{E}\left[\mathbf{Y}_i \,|\, \mathbf{Y}_1 = b_1, \ldots, \mathbf{Y}_{i-1} = b_{i-1}\right] \ge \delta/4 \tag{15}$$

for all i and all $b_1, \ldots, b_{i-1} \in \{0,1\}$. Let $\mathbf{X}_0 \equiv 0$ and

$$\mathbf{X}_i = \mathbf{X}_{i-1} + \mathbf{Y}_i - \mathbf{E}\left[\mathbf{Y}_i \,|\, \mathbf{Y}_1, \cdots, \mathbf{Y}_{i-1}\right].$$

Then $\mathbf{X}_0, \mathbf{X}_1, \ldots$ is a martingale that satisfies $|\mathbf{X}_i - \mathbf{X}_{i-1}| \le 1$ with probability 1, and we have that

$$\mathbf{X}_{32\delta^2 n} = \sum_{i=1}^{32\delta^2 n} \left(\mathbf{Y}_i - \mathbf{E}\left[\mathbf{Y}_i \,|\, \mathbf{Y}_1, \cdots, \mathbf{Y}_{i-1}\right]\right) \le \sum_{i=1}^{32\delta^2 n} \mathbf{Y}_i - 8\delta^3 n,$$

using (15) for the inequality. Applying the Azuma-Hoeffding inequality (see, e.g., Theorem 5.1 of [DP09]) to the martingale $\mathbf{X}_0, \mathbf{X}_1, \ldots$, we get that

$$\mathbf{Pr}\left[\sum_i \mathbf{Y}_i \le 2\delta^3 n\right] \le \mathbf{Pr}\left[\mathbf{X}_{32\delta^2 n} \le 2\delta^3 n - 8\delta^3 n\right] \le \exp\left(-\frac{(6\delta^3 n)^2}{2 \cdot 32\delta^2 n}\right) < \exp(-n\delta^4/2).$$

This finishes the proof of the claim. □

We are almost done with the proof of Lemma 2.7. Recalling that $\delta = 7^{\ell-2}\varepsilon$, we have that

$$\exp\left(-n\delta^4/2\right) \le \delta/4 \quad \text{and} \quad \delta \le 2^{-8} \tag{16}$$

since $d \ge 2$, $n \ge 2^{60d}$, $\varepsilon = 2^{-12d}$ and $\ell \in \{2, \ldots, d\}$. It follows from Claim 2.9 that

$$\begin{aligned} \mathbf{E}\left[\mathbf{\Gamma}\right] &\le \exp(-n\delta^4/2) \cdot 2n + (1 - \exp(-n\delta^4/2)) \cdot \left(2 \cdot 32\delta^2 n + (1+6\delta) \cdot (n - 32\delta^2 n)\right) \\ &< \delta n/2 + 64\delta^2 n + (1+6\delta)n \le (1+7\delta)n, \end{aligned}$$

where we also used the two inequalities in (16). This finishes the proof of the lemma. □

2.4 Proof of Theorem 1.

Finally we combine all the ingredients to prove Theorem 1.

Recall that $d \ge 2$, n and N are positive integers that satisfy $n \ge 2^{60d}$ and $N \ge (2^{13}n)^d \ge N_d$. We also have $\varepsilon = 2^{-12d}$ and $M = 2^{\varepsilon^5 n}$. We first prove by induction on ℓ that, for $\ell = 1, \ldots, d$, any

monotone majority circuit F over $\{0,1\}^{(\ell+1)\times N_\ell}$ of depth ℓ and size at most M satisfies

$$\begin{aligned}&\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}_\ell}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}_\ell}\left[F(\boldsymbol{y})=0\right]\le 1+7^{\ell-1}\varepsilon,\quad\text{and}\\&\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}'_\ell}\left[F(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}'_\ell}\left[F(\boldsymbol{y})=0\right]\le 1+7^{\ell-1}\varepsilon.\end{aligned}\tag{17}$$

The $\ell=1$ base case follows from Lemma 2.1. Now assume that (17) holds for $\ell-1$. By Lemma 2.7, any monotone majority circuit F^* over $\{0,1\}^{\ell\times N^*_\ell}$ of depth ℓ and size at most M satisfies

$$\mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{YES}^*_\ell}\left[F^*(\boldsymbol{x})=1\right]+\mathbf{Pr}_{\boldsymbol{y}\sim\mathcal{NO}^*_\ell}\left[F^*(\boldsymbol{y})=0\right]\le 1+7^{\ell-1}\varepsilon.\tag{18}$$

It follows from Lemmas 2.5 and 2.6 that every monotone majority circuit F over $\{0,1\}^{(\ell+1)\times N_\ell}$ of depth ℓ and size at most M satisfies (17). This finishes the induction.

We finish the proof using $(\mathcal{YES}^*_d,\mathcal{NO}^*_d)$ over $\{0,1\}^{d\times N^*_d}$, where $N^*_d=N_d-1<N$. Given (18) on $(\mathcal{YES}^*_d,\mathcal{NO}^*_d)$ and the fact that $7^{d-1}\varepsilon<1$, no depth-d monotone majority circuit on $\{0,1\}^{d\times N^*_d}$ of size at most M can compute U_{d,N^*_d} correctly on all inputs, because every string $\boldsymbol{x}\sim\mathcal{YES}^*_d$ has $\mathsf{SUM}(\boldsymbol{x})=2^{N^*_d}$ and hence $U_{d,N^*_d}(\boldsymbol{x})=1$, while every string $\boldsymbol{y}\sim\mathcal{NO}^*_d$ has $\mathsf{SUM}(\boldsymbol{y})=2^{N^*_d}-d$ and hence $U_{d,N^*_d}(\boldsymbol{y})=0$. Since $N>N^*_d$, this establishes Theorem 1.

3 The Upper Bound: Proof of Theorem 2.

We prove Theorem 2 in this section. We focus on the case when $N^{1/d}>1$ is a positive integer, and give a depth-d monotone majority circuit that computes $U_{k,N}$ and has size at most

$$2^{3(N^{1/d}\cdot\log k+\log N)}.\tag{19}$$

For the general case, we let $n=\lceil N^{1/d}\rceil>1$, and let s denote the smallest integer such that $n^s\ge N$ (so $s\le d$). Then we first construct a depth-s monotone majority circuit that computes U_{k,n^s}, and then hard-wire the variables in the last n^s-N columns to be 0 to get a circuit for $U_{k,N}$. The size bound given in the statement of Theorem 2 follows from (19) and the simple facts that $n\le 2N^{1/d}$ and $n^s\le nN\le N^2$. For the rest of the section we assume that $n=N^{1/d}>1$ is an integer.

First we note that the theorem (with the size bound as given in (19); the same below) is trivial if $N<\log k$ since implementing $U_{k,N}$ directly using a single THR gate only takes a total weight of $k\cdot 2^N<2^{3\log k}$. Assuming that $N\ge\log k$ below, we let $t\in\{1,\dots,d\}$ denote the smallest integer such that $n^t=N^{t/d}\ge\log k$. We also write $M=n^t$. It is clear by the choice of t that we have

$$M\le n\log k.\tag{20}$$

With the same reasoning the theorem is trivial if $M=N$. Below we assume that $t\le d-1$.

We need some notation for our construction. We say $\mathcal{S}=(S_1,\dots,S_\ell)$ is an *ℓ-decomposition* of $[N]$ if there exist indices $1=a^-_1\le a^+_1<a^-_2\le a^+_2<\ldots<a^-_\ell\le a^+_\ell=N$ such that

- For each $\gamma\in[\ell]$, $S_\gamma=\{a^-_\gamma,a^-_\gamma+1,\dots,a^+_\gamma\}$; and
- $\bigcup_{\gamma\in[\ell]}S_\gamma=[N]$.

In other words, $\mathcal{S}$ partitions $[N]$ into ℓ sequential intervals.

Let $(x_{i,j})_{i\in[k],j\in[N]}$ be the set of input variables of $U_{k,N}$. Given an ℓ-decomposition $\mathcal{S}$, we define a sequence of "conditional" *carry-bit functions* $c^{(\gamma)}_{\alpha,\beta}(x)$, where $0 \le \alpha, \beta \le k-1$ and $\gamma \in [\ell]$. Each function $c^{(\gamma)}_{\alpha,\beta}$ depends only on the variables $x_{i,j}$ with $j \in S_\gamma$. For convenience, let $B_\gamma = [k] \times S_\gamma$ be the set containing the indices of these variables. Intuitively, for an assignment $x \in \{0,1\}^{k\times N}$, we have $c^{(\gamma)}_{\alpha,\beta}(x) = 1$ if and only if a carry of value at least α is generated/propagated by the input bits corresponding to B_γ, assuming this block of variables receives a carry of value β from the block of variables to the right. Formally,

$$c^{(\gamma)}_{\alpha,\beta}(x) \stackrel{\text{def}}{=} 1 \quad \Longleftrightarrow \quad \sum_{(i,j)\in B_\gamma} 2^{|S_\gamma|-(j+1-a^-_\gamma)} \cdot x_{i,j} + \beta \ge \alpha \cdot 2^{|S_\gamma|}. \tag{21}$$

For each $i \in \{0,\dots,d-t\}$, we write $\mathcal{S}^{(i)}$ to denote the n^i-decomposition in which each set has size n^{d-i}. Observe that, for the 1-decomposition $\mathcal{S}^{(0)} = (S^{(0)}_1)$, where $S^{(0)}_1 = \{1,\dots,N\}$, we have

$$U_{k,N}(x) = 1 \quad \Longleftrightarrow \quad c^{(1)}_{1,0}(x) = 1, \tag{22}$$

for the function $c^{(1)}_{1,0}$ of $\mathcal{S}^{(0)}$.

Our construction is based on a recursive computation of functions $c^{(\gamma)}_{\alpha,\beta}(\cdot)$ associated to different decompositions $\mathcal{S}^{(r)}$, for r from $d-t$ back to 0, where each decomposition $\mathcal{S}^{(r+1)}$ is obtained via a refinement of the previous decomposition $\mathcal{S}^{(r)}$. More precisely we construct our monotone majority circuit for $U_{k,N}$ with the following intended behavior. The top gate of the circuit computes the bit $c^{(1)}_{1,0}(x)$ associated to the decomposition $\mathcal{S}^{(0)}$. However, this gate does not have access to x: it receives as input the output of carry-bit functions $c^{(\gamma)}_{\alpha,\beta}(x)$ corresponding to the finer n-decomposition $\mathcal{S}^{(1)}$ in which each block has n^{d-1} columns. This then leads to a recursive procedure, which unfolds as a depth-$(d-t+1)$ circuit described in more detail below (recall that $t \ge 1$).

In general our circuit has $d-t+1$ layers of majority gates, where gates at the ith layer compute carry-bit functions $c^{(\ell)}_{\alpha,\beta}$ corresponding to the $n^{d-t-i+1}$-decomposition $\mathcal{S}^{(d-t-i+1)}$. The base case, i.e. the first layer of majority gates that are supposed to compute $c^{(\ell)}_{\alpha,\beta}$ of $\mathcal{S}^{d-t}$, is done by a majority gate that follows directly the definition given in (21). It is clear that the size of each gate in the first layer is bounded from above by $k2^M$.

Due to the recursive nature of our construction, it is sufficient to describe how to compute the carry-bit functions corresponding to a decomposition $\mathcal{S}^{(r)}$ from the carry-bit functions corresponding to $\mathcal{S}^{(r+1)}$ for each $r \in \{0,1,\dots,d-t-1\}$. For convenience we fix an r below and write $\mathcal{S}'$ for $\mathcal{S}^{(r)}$ and $\mathcal{S}$ for $\mathcal{S}^{(r+1)}$. We also fix a set $S' \in \mathcal{S}'$ with $S' = S_1 \cup \ldots \cup S_n$, where $S_1,\dots,S_n$ are sets in the ordered tuple $\mathcal{S}$ listed from left to right. We write $c_{u,v}$ to denote a carry-bit function of the block S that we need to compute, for some $u,v \in \{0,\dots,k-1\}$, and assume that we have already computed $c^{(\gamma)}_{\alpha,\beta}$ for each block S_γ, $\gamma \in [n]$, and for all $\alpha,\beta \in \{0,\dots,k-1\}$. The goal is to compute $c_{u,v}(x)$ given the bits $c^{(\gamma)}_{\alpha,\beta}(x)$.

We start with a general observation about carry-bit functions of a block. We say that $(\alpha,\beta) \prec (\alpha',\beta')$ if either $\alpha < \alpha'$, or $\alpha = \alpha'$ and $\beta \ge \beta'$. Given a block S_γ, note that $c^{(\gamma)}_{\alpha,\beta}$ has the following monotonicity property. (Note that the assumption of $|S_\gamma| \ge \log k$ always holds given our choice of t and trivial cases ruled out at the beginning of the section.)

Claim 3.1. *Assume that $|S_\gamma| \ge \log k$. If $(\alpha,\beta) \prec (\alpha',\beta')$, then $c^{(\gamma)}_{\alpha,\beta}(x) \ge c^{(\gamma)}_{\alpha',\beta'}(x)$ on every input string x for $U_{k,N}$.*

Proof. We consider the two cases corresponding to the assumption that $(\alpha, \beta) \prec (\alpha', \beta')$. If $\alpha = \alpha'$ and $\beta \geq \beta'$, the claim follows immediately from (21).

Assume now that $\alpha < \alpha'$, where $\beta, \beta' \in \{0, \ldots, k-1\}$ are arbitrary. Clearly it suffices to argue that $c^{(\gamma)}_{\alpha', k-1}(x) = 1$ implies that $c^{(\gamma)}_{\alpha'-1,0}(x) = 1$. Using (21), this assumption is equivalent to

$$\sum_{(i,j) \in B_\gamma} 2^{|S_\gamma| - (j+1-a^-_\gamma)} \cdot x_{i,j} + (k-1) \geq \alpha' \cdot 2^{|S_\gamma|}. \tag{23}$$

In order to show $c^{(\gamma)}_{\alpha'-1,0}(x) = 1$, we need to verify that

$$\sum_{(i,j) \in B_\gamma} 2^{|S_\gamma| - (j+1-a^-_\gamma)} \cdot x_{i,j} \geq (\alpha' - 1) \cdot 2^{|S_\gamma|}.$$

Using (23) it is sufficient to have $k - 1 \leq 2^{|S_\gamma|}$. This follows from the assumption in the statement of the claim, which completes the proof. □

The description of the majority gate that computes $c_{u,v}(x)$ for the block S' in $\mathcal{S}'$ using $c^{(\gamma)}_{\alpha,\beta}(x)$ for blocks $S_1, \ldots, S_n$ in $\mathcal{S}$ is based on the following lemma.

Lemma 3.2. *Assume that $|S_\gamma| \geq \log k$ for every $\gamma \in [n]$. Then $c_{u,v}(x) = 1$ if and only if*

$$v + \sum_{\gamma=1}^{n} \left(\left(\sum_{\alpha=1,\beta=0}^{k-1} c^{(\gamma)}_{\alpha,\beta}(x) \right) \cdot k^{n-\gamma} \right) \geq u \cdot k^n. \tag{24}$$

Proof. We consider (24) as a sum in base k over $k(k-1)$ rows and n columns of variables, with v extra 1's on column n (which corresponds to the least significant position). Let p_γ denote the (base k) carry from column γ to column $\gamma - 1$ in (24), and let q_γ denote the (base 2) carry from block γ to block $\gamma - 1$ in our decomposition of x after adding v to block n (without taking into account the remaining columns of x not covered by $S_1 \cup \ldots \cup S_n$).

We prove by induction that $p_\gamma = q_\gamma$, for all γ from n to 1. Notice that this establishes the lemma. For the basis when $\gamma = n$, we consider the following two cases:

1. If $c^{(n)}_{\alpha,\beta} = 0$ for all $\alpha \geq 1$ and $\beta \geq 0$, then $q_n = 0$ (as we have $c^{(n)}_{1,k-1} = 0$ and $v \leq k-1$). This implies that $p_n = q_n = 0$.

2. Otherwise, let (α_n, β_n) denote the largest pair (under $\prec$ defined earlier) with $c^{(n)}_{\alpha_n,\beta_n} = 1$. It follows from Claim 3.1 that $q_n = \alpha_n$ if $\beta_n \leq v$, and $q_n = \alpha_n - 1$ if $\beta_n > v$. We also have

$$v + \sum_{\alpha=1,\beta=0}^{k-1} c^{(n)}_{\alpha,\beta} = (\alpha_n - 1) \cdot k + (k - \beta_n + v).$$

It follows from this equation and the characterization of q_n that the (base k) carry $p_n = q_n$.

The induction step is similar. We assume that $p_{\gamma+1} = q_{\gamma+1}$, and prove that $p_\gamma = q_\gamma$. We focus on the γ-th column from (24) and block γ, and consider the following two cases:

1. If $c^{(n)}_{\alpha,\beta} = 0$ for all $\alpha \geq 1$ and $\beta \geq 0$, then $q_\gamma = 0$ (as we have $c^{(\gamma)}_{1,k-1} = 0$ and $q_{\gamma+1} \leq k-1$). This implies that $p_\gamma = q_\gamma = 0$.

2. Otherwise, let $(\alpha_\gamma, \beta_\gamma)$ denote the largest pair with $c^{(\gamma)}_{\alpha_\gamma,\beta_\gamma} = 1$. Using Claim 3.1, q_γ is α_γ if $\beta_\gamma \leq q_{\gamma+1}$, and q_γ is $\alpha_\gamma - 1$ if $\beta_\gamma > q_{\gamma+1}$. Using the inductive hypothesis, we have

$$p_{\gamma+1} + \sum_{\alpha=1,\beta=0}^{k-1} c^{(n)}_{\alpha,\beta} = (\alpha_\gamma - 1) \cdot k + (k - \beta_\gamma + q_{\gamma+1}).$$

   It follows from this equation and the characterization of q_γ that $p_\gamma = q_\gamma$.

This finishes the induction, and the proof of the lemma. □

Lemma 3.2, (21), and our previous discussions complete the description of the circuit for $U_{k,N}$. Moreover, its correctness follows easily from (22) and Lemma 3.2. It remains to analyze the size of the resulting depth-$(d-t+1)$ majority circuit.

We upper bound its size layer by layer as follows. As discussed earlier, the size of each majority gate in the first layer is at most $k2^M$, and there are n^{d-t} many of them. Furthermore, for the i-th layer of the circuit, where $i > 1$, there are $n^{(d-t-i+1)}$ gates each of which has size at most

$$k(k-1) \cdot \frac{k^n - 1}{k-1} < k^{n+1},$$

as given in Lemma 3.2. Using (20), the majority circuit for $U_{k,N}$ has overall size at most

$$n^{d-t} \cdot k2^M + \sum_{i=2}^{d-t+1} n^{d-t+1-i} \cdot k^{n+1} \leq Nk2^M + 2Nk^{n+1} \leq 2^{3(N^{1/d} \log k + \log N)}.$$

The construction presented here uses THR gates and majority circuits. We sketch in Appendix A an alternative construction with respect to semi-unbounded fan-in AND/OR circuits.

4 Strengthening the Ajtai-Gurevich Result: Proof of Theorem 3.

We require the following lemma:

Lemma 4.1. *For a suitable absolute constant $0 < c < 1$, letting $k = (\log N)^c$, the function $U_{k,N}$ is computed by a poly(N)-size* AND/OR/NOT *circuit of depth* 3.

Proof. Recall the well-known technique of carry-save addition, also known as the "3-to-2 trick," for addition of binary numbers (see e.g., Section 1.2.3 of [Lei92]). This "trick" states that there is a (multi-output) circuit that takes as input three n-bit binary numbers X, Y, Z and outputs two $(n+1)$-bit binary numbers A, B such that (i) $A + B = X + Y + Z$, and (ii) each output bit A_i or B_i depends on at most 3 of the input bits. By applying this trick in parallel to the N-bit integers $x^{(1)}, \ldots, x^{(k)}$ that are the rows of the input to $U_{k,N}$, we obtain $\lceil 2k/3 \rceil$ many $(N+1)$-bit integers whose sum equals $x^{(1)} + \cdots + x^{(k)}$. Recursing $O(\log k)$ times, we see that there are two $(N + O(\log k))$-bit integers (call them y and z) such that $y + z = x^{(1)} + \cdots + x^{(k)}$. A naive composition of these "3-to-2 trick" circuits in a tree of depth $O(\log k)$ to compute y, z would yield

a circuit of depth $\Theta(\log \log N)$. To avoid this blowup in circuit depth, we proceed differently, by observing that each each bit y_i, z_i depends on at most $3^{O(\log k)} \leq \log N$ of the original input bits of the $x^{(i)}$'s, and exploiting this locality to get a depth-3 circuit overall.

In more detail, let y_i denote the bit in the "2^i-position" of the binary representation of y, so

$$y = \sum_{i=0}^{N+O(\log k)} y_i \cdot 2^i \quad \text{and similarly} \quad z = \sum_{i=0}^{N+O(\log k)} z_i \cdot 2^i.$$

We define "generate" and "propagate" bits for each bit position of $y + z$ in the standard way,

$$g_i \stackrel{\text{def}}{=} y_i \wedge z_i \quad \text{and} \quad p_i \stackrel{\text{def}}{=} y_i \vee z_i,$$

so $g_i = 1$ iff the bits in the 2^i-position generate a carry into the 2^{i+1}-position, and $p_i = 1$ iff the bits in the 2^i-position propagate an incoming carry into the 2^i-position onward to the 2^{i+1}-position. Observe that each p_i, g_i depends on at most $2 \log N$ of the original input bits.

The sum $y + z$ is at least 2^N if and only if either of the following events hold:

- Event A: at least one of the bits $y_N, y_{N+1}, \ldots, z_N, z_{N+1}, \ldots$ is 1. This can be expressed as

  $$A = \bigvee_{j \geq N} (y_j \vee z_j).$$

  Since y_j, z_j each depend on at most $\log N$ of the original input variables, each of them can be expressed as a poly(N)-size DNF over the original input variables, and thus A can be expressed as a poly(N)-size DNF.

- Event B: a carry bit is propagated into the 2^N-position. Event B can be expressed as

  $$B = \bigvee_{j=1}^{N-1} \left(g_j \wedge \left(\bigwedge_{j<i<N} p_i \right) \right).$$

  As each p_i depends on at most $2 \log N$ of the original input variables, it can be expressed as a poly(N)-size CNF; the same holds for g_j, so

  $$\left(g_j \wedge \left(\bigwedge_{j<i<N} p_i \right) \right)$$

  can be expressed as a poly(N)-size CNF, and thus Event B can be expressed as a poly(N)-size depth-3 OR-AND-OR circuit.

As a consequence, $A \vee B$ can be expressed as a poly(N)-size depth-3 OR-AND-OR circuit over the original input variables, and the lemma is proved. □

Proof of Theorem 3: We take $g_N \stackrel{\text{def}}{=} U_{k,N}$, where $k = (\log N)^c$ as in Lemma 4.1. Then Part (i) of the theorem follows from Lemma 4.1. Part (ii) follows from our main lower bound, Theorem 1, by

observing that any circuit for $U_{k,N}$ yields a circuit for $U_{d,N}$ (by setting the last $k-d$ rows of the input to 0). □

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987. 1

[ABFR94] James Aspnes, Richard Beigel, Merrick L. Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994. 1

[AG87] Miklós Ajtai and Yuri Gurevich. Monotone versus positive. *J. ACM*, 34(4):1004–1015, 1987. 1, 1.1

[All89] Eric Allender. A note on the power of threshold circuits. In *Symposium on Foundations of Computer Science* (FOCS), pages 580–584, 1989. 1

[AM05] Kazuyuki Amano and Akira Maruoka. On the complexity of depth-2 circuits with threshold gates. In *Symposium on Mathematical Foundations of Computer Science* (MFCS), pages 107–118, 2005. 1

[And85] Alexander E. Andreev. On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Math. Dokl*, 31(3):530–534, 1985. 1

[BHKS14] Olaf Beyersdorff, Edward A. Hirsch, Jan Krajícek, and Rahul Santhanam. Optimal algorithms and proofs (Dagstuhl Seminar 14421). *Dagstuhl Reports*, 4(10):51–68, 2014. 1, 1.1

[BRS95] Richard Beigel, Nick Reingold, and Daniel A. Spielman. PP is closed under intersection. *J. Comput. Syst. Sci.*, 50(2):191–202, 1995. 1

[BST13] Eric Blais, Dominik Scheder, and Li-Yang Tan. Ajtai-Gurevich Redux. Manuscript, 2013. 1, 1.1

[BW05] Amos Beimel and Enav Weinreb. Monotone circuits for weighted threshold functions. In *Conference on Computational Complexity* (CCC), pages 67–75, 2005. 2, A

[Der65] Michael Dertouzos. *Threshold Logic: A Synthesis Approach*. MIT Press, 1965. 1

[DP09] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009. 2.3

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. 1

[GHR92] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992. 1

[GK93] Mikael Goldmann and Marek Karpinski. Simulating threshold circuits by majority circuits. In *Symposium on Theory of Computing* (STOC), pages 551–560. ACM, 1993. 1, 1.1

[Hås10] Johan Håstad. Some Results in Circuit Complexity. Presentation at *China Theory Week* (CTW). Slides available at: http://conference.itcs.tsinghua.edu.cn/CTW2010/content/Slides/1.pdf, 2010. 1, 1.1

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991. 1.2

[Hof92] Thomas Hofmeister. The power of negative thinking in constructing threshold circuits for addition. In *Structure in Complexity Theory Conference* (CCC), pages 20–26, 1992. 1, 1.1

[Hof96] Thomas Hofmeister. A note on the simulation of exponential threshold weights. In *Conference on Computing and Combinatorics* (COCOON), pages 136–141, 1996. 1

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*. Springer, 2012. 1

[Lei92] Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992. 4

[MP68] Marvin Minsky and Seymour Papert. *Perceptrons - An Introduction to Computational Geometry*. MIT Press, 1968. 1

[Mur71] Saburo Muroga. *Threshold Logic and its Applications*. Wiley, 1971. 1

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004. 1

[Par94] Ian Parberry. *Circuit Complexity and Neural Networks*. MIT Press, 1994. 1

[Raz85] Alexander A. Razborov. Lower bounds for the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady*, 31(6):354–357, 1985. 1

[SB91] Kai-Yeung Siu and Jehoshua Bruck. On the power of threshold circuits with small weights. *SIAM J. Discrete Math.*, 4(3):423–435, 1991. 1

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. 1

[She07] Alexander A. Sherstov. Powering requires threshold depth 3. *Inf. Process. Lett.*, 102(2-3):104–107, 2007. 1

[Sto95] Alexei P. Stolboushkin. Finitely monotone properties. In *Symposium on Logic in Computer Science* (LICS), pages 324–330, 1995. 1

[Tar88] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. 1

[TZ92] Alan Taylor and William Zwicker. A characterization of weighted voting. *Proc. Amer. Math. Soc.*, 115(4):1089–1094, 1992. 1

[Yao89] Andrew Chi-Chih Yao. Circuits and local computation. In *Symposium on Theory of Computing* (STOC), pages 186–196, 1989. 1.2

[Yao90] Andrew Chi-Chih Yao. On ACC and threshold circuits. In *Symposium on Foundations of Computer Science* (FOCS), pages 619–627, 1990. 1

A Upper Bound for the Universal Monotone Threshold Gate.

We sketch in this section a construction of monotone circuits for the universal monotone threshold function that matches the parameters obtained by Beimel and Weinreb [BW05]. More precisely, we describe a polynomial size $O(\log N)$-depth AND/OR circuit for $U_{O(N),O(N \log N)}$, where OR gates have unbounded fan-in, while AND gates have fan-in two.

Our construction relies on a more general reduction from $U_{k,N}$ to a certain graph connectivity problem. We start with an ℓ-decomposition $\mathcal{S}$ of $U_{k,N}$ (see Section 3 for more details), and assume (for now) that we are given the corresponding (conditional) carry-bit functions $c^{(\gamma)}_{\alpha,\beta}(x)$, where α and β are in $\{0,\ldots,k-1\}$, and $\gamma \in [\ell]$.

Given these bits, we can view them as a layered directed graph $G_{\mathcal{S},x} = (V, E)$ which depends on x and $\mathcal{S}$ as follows. The vertices of G are partitioned into $\ell + 1$ layers, which we number for convenience from ℓ to 0. The first and last layers are special, and contain a single vertex only. The remaining layers each contain k vertices. The (directed) edges of this graph leave the γ-th layer and reach the $(\gamma-1)$-th layer. We use the output bit of each function $c^{(\gamma)}_{\alpha,\beta}$ to decide whether an edge is present in this graph. The idea is that there will be a path from the ℓ-th layer to the 0-th layer if and only if $U_{k,N}(x) = 1$.

More precisely, we view $V = L_\ell \cup L_{\ell-1} \cup \ldots \cup L_0$, where $L_\ell = \{s\}$, $L_0 = \{t\}$, and $L_\gamma = \{v_{\gamma,0},\ldots,v_{\gamma,k-1}\}$, for $\ell > \gamma > 0$. The edge set $E \subseteq V \times V$ is defined as follows.

- $(s, v_{\ell-1,j}) \in E$ if and only if $c^{(\ell)}_{1,j} = 1$, where $j \in \{0,\ldots,k-1\}$;
- $(v_{1,j}, t) \in E$ if and only if $c^{(1)}_{j,0} = 1$, where $j \in \{0,\ldots,k-1\}$;
- For $\ell - 1 \geq \gamma \geq 2$ and $0 \leq \alpha, \beta \leq k-1$, $(v_{\gamma,\alpha}, v_{\gamma-1,\beta}) \in E$ if and only if $c^{(\gamma)}_{\alpha,\beta} = 1$;
- There is no other edge in E.

Given vertices u, v in a graph G, we write $u \rightsquigarrow v$ if there exists a directed path from u to v in G. Our construction is based on the following observation.

Lemma A.1. *Given an ℓ-decomposition $\mathcal{S}$ for $U_{k,N}$ and an input x,*

$$U_{k,N}(x) = 1 \quad \Longleftrightarrow \quad s \rightsquigarrow t \text{ in } G_{\mathcal{S},x}.$$

Proof. We provide a sketch of the argument. If $U_{k,N}(x) = 1$, consider the sequence of carries generated during the actual computation of $\sum_{i\in[k]} x^{(i)}$ by the standard binary addition algorithm. At least one final carry is generated in this process, since the sum is at least 2^N. The correct carry values computed during intermediate steps of the addition algorithm correspond to a path from s to t in $G_{\mathcal{S},x}$. On the other hand, if there exists a path from s to t in this graph, then an inductive argument starting from t and proceeding backwards to s shows that, during each step of the addition algorithm, at least some number of carries must be produced when we add the integers $x^{(1)}, \ldots, x^{(k)}$. In particular, there must be at least one final carry bit, which implies that $U_{k,N}(x) = 1$. □

To sum up, in order to compute $U_{k,N}$ from the carry-bit functions it is enough to solve a directed s-t-connectivity problem on a graph with $O(N)$ layers, where each layer contains $O(k)$ vertices.

The computation of the carry-bit functions can be done efficiently in the case of the universal monotone threshold function if we start with an $\Omega(N \log N)$-decomposition. More precisely, each such function can be written as a monotone majority gate over a polynomial number of input bits, which is known to admit efficient monotone circuits as needed in our construction.

Finally, the upper bound follows from the well-known construction of monotone circuits for s-t-connectivity on layered graphs via divide-and-conquer.